



Faster Inversion and Other Black Box Matrix Computations Using Efficient Block Projections

Wayne Eberly, Mark Giesbrecht, Pascal Giorgi, Arne Storjohann, Gilles Villard

► To cite this version:

Wayne Eberly, Mark Giesbrecht, Pascal Giorgi, Arne Storjohann, Gilles Villard. Faster Inversion and Other Black Box Matrix Computations Using Efficient Block Projections. 2007. hal-00127807

HAL Id: hal-00127807

<https://hal.science/hal-00127807>

Preprint submitted on 29 Jan 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Faster Inversion and Other Black Box Matrix Computations Using Efficient Block Projections

Wayne Eberly¹, Mark Giesbrecht², Pascal Giorgi^{2,4}, Arne Storjohann², Gilles Villard³

(1) Department of Computer Science, U. Calgary
<http://pages.cpsc.ucalgary.ca/~eberly>

(2) David R. Cheriton School of Computer Science, U. Waterloo
<http://www.uwaterloo.ca/~{mwg,pgiorgi,astorjoh}>

(3) CNRS, LIP, École Normale Supérieure de Lyon
<http://perso.ens-lyon.fr/gilles.villard>

(4) IUT - Université de Perpignan
<http://webdali.univ-perp.fr/~pgiorgi>

ABSTRACT

Efficient block projections of non-singular matrices have recently been used, in [11], to obtain an efficient algorithm to find rational solutions for sparse systems of linear equations. In particular a bound of $O(n^{2.5})$ machine operations is obtained for this computation assuming that the input matrix can be multiplied by a vector with constant-sized entries using $O(n)$ machine operations. Somewhat more general bounds for black-box matrix computations are also derived. Unfortunately, the correctness of this algorithm depends on the existence of efficient block projections of non-singular matrices and this has been conjectured but not proved.

In this paper we establish the correctness of the algorithm from [11] by proving the existence of efficient block projections for arbitrary non-singular matrices over sufficiently large fields. We further demonstrate the usefulness of these projections by incorporating them into existing black-box matrix algorithms to derive improved bounds for the cost of several matrix problems — considering, in particular, “sparse” matrices that can be multiplied by a vector using $O(n)$ field operations: We show how to compute the dense inverse of a sparse matrix over any field using an expected number of $O(n^{2.27})$ operations in that field. A basis for the null space of a sparse matrix — and a certification of its rank — are obtained at the same cost. An application of this technique to Kaltofen and Villard’s Baby-Steps/Giant-Steps algorithms for the determinant and Smith Form of an

*This material is based on work supported in part by the French National Research Agency (ANR Gecko, Villard), and by the Natural Sciences and Engineering Research Council (NSERC) of Canada (Eberly, Giesbrecht, Storjohann).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

integer matrix is also sketched, yielding algorithms requiring $O(n^{2.66})$ machine operations. More general bounds involving the number of black-box matrix operations to be used are also obtained.

The derived algorithms are all probabilistic of the Las Vegas type. That is, they are assumed to be able to generate random elements from the field at unit cost, and always output the correct answer in the expected time given.

1. INTRODUCTION

In our paper [11] we presented an algorithm which purportedly solved a sparse system of rational equations considerably more efficiently than standard linear equations solving. Unfortunately, its effectiveness in all cases was conjectural, even as its complexity and actual performance were very appealing. This effectiveness relied on a conjecture regarding the existence of so-called efficient block projections. Given a matrix $A \in \mathbb{F}^{n \times n}$ over any field \mathbb{F} , these projections should be block vectors $u \in \mathbb{F}^{n \times s}$ (where s is a blocking factor dividing n , so $n = ms$) such that we can compute uv or $v^t u$ quickly for any $v \in \mathbb{F}^{n \times s}$, and such that the sequence of vectors $u, Au, \dots, A^{m-1}u$ has rank n .

In this paper, we prove the existence of a class of such efficient block projections for non-singular $n \times n$ matrices over sufficiently large fields — we require that the size of the field \mathbb{F} exceed $n(n+1)$. This implies our algorithm from [11] for finding the solution to $A^{-1}b$ for a “sparse” system of equations $A \in \mathbb{Z}^{n \times n}$ and $b \in \mathbb{Z}^{n \times 1}$ works as stated using these projections, and requires fewer bit operations than any previously known when A is sparse, at least using “standard” (i.e., cubic) matrix arithmetic. Here, by A being sparse, we mean that it has a fast matrix-vector product modulo any small (machine-word size) prime p . In particular, our algorithm requires an expected $O(n^{1.5}(\log(\|A\| + \|b\|)))$ matrix-vector products $v \mapsto Av \bmod p$ for $v \in \mathbb{Z}_p^{n \times 1}$ plus and additional $O(n^{2.5}(\log(\|A\| + \log \|b\|)))$ bit operations. The algorithm is probabilistic of the Las Vegas type. That is, it assumes the ability to generate random bits at unit cost, and always returns the correct answer with controllably high probability. When $\phi(n) = O(n)$, the implied cost of $O(n^{2.5})$ bit operations improves upon the p -adic lift-

ing method of [7] which requires $O(n^3)$ bit operations for sparse or dense matrices. This theoretical efficiency was reflected in practice in [11] at least for large matrices.

We present several other rather surprising applications of this technique. Each incorporates the technique into an existing algorithm in order to reduce the asymptotic complexity for the sparse matrix problem to be solved. In particular, given a matrix $A \in \mathbb{F}^{n \times n}$ over an arbitrary field \mathbb{F} , we are able to compute the complete inverse of A with $O(n^{3-1/(\omega-1)})$ operations in \mathbb{F} plus $O(n^{2-1/(\omega-1)})$ matrix-vector products by A . Here ω is such that we can multiply two $n \times n$ matrices with $O(n^\omega)$ operations in \mathbb{F} . Standard matrix multiplication gives $\omega = 3$, while the best known matrix multiplication of Coppersmith & Winograd [6] has $\omega = 2.376$. If again we can compute $v \mapsto Av$ with $O(n)$ operations in \mathbb{F} , this implies an algorithm to compute the inverse with $O(n^{3-1/(\omega-1)})$ operations in \mathbb{F} . This is always less than $O(n^\omega)$, and in particular equals $O(n^{2.27})$ operations in \mathbb{F} for the best known ω of [6]. Other relatively straightforward applications of these techniques yield algorithms for the full nullspace and (certified) rank with this same cost. Finally, we sketch how these methods can be employed in the algorithms of Kaltofen and Villard [20] and [14] to computing the determinant and Smith form of sparse matrices more efficiently.

There has certainly been much important work done on finding exact solutions to sparse rational systems prior to [11]. Dixon's p -adic lifting algorithm [7] performs extremely well in practice for dense and sparse linear systems, and is implemented efficiently in LinBox [8], Maple and Magma (see [11] for a comparison). Kaltofen & Saunders [18] are the first to propose to use Krylov-type algorithms for these problems. Krylov-type methods are used to find Smith forms of sparse matrices and to solve Diophantine systems in parallel in [13, 14], and this is further developed in [20, 9]. See the references in these papers for a more complete history. For sparse systems over a field, the seminal work is that of Wiedemann [23] who shows how to solve sparse $n \times n$ systems over a field with $O(n)$ matrix-vector products and $O(n^2)$ other operations. This research is further developed in [18, 5, 17] and many other works. The bit complexity of similar operations for various families of structure matrices is examined in [12].

2. SPARSE BLOCK GENERATORS FOR THE KRYLOV SPACE

For now we will consider an arbitrary invertible matrix $A \in \mathbb{F}^{n \times n}$ over a field \mathbb{F} , and s an integer — the blocking factor — which divides n exactly. Let $m = n/s$. For a so-called *block projection* $u \in \mathbb{F}^{n \times s}$ and $1 \leq k \leq m$, we denote by $\mathcal{K}_k(A, u)$ the *block Krylov matrix* $[u, Au, \dots, A^{k-1}u] \in \mathbb{F}^{n \times ks}$. Our goal is to show that $\mathcal{K}_m(A, u) \in \mathbb{F}^{n \times n}$ is nonsingular for a particularly simple and sparse u , assuming some properties of A .

Our factorization uses the special projection (which we will refer to as an *efficient block projection*)

$$u = \begin{bmatrix} I_s \\ \vdots \\ I_s \end{bmatrix} \in \mathbb{F}^{n \times s} \quad (2.1)$$

which is comprised of m copies of I_s and thus has exactly n non-zero entries. A similar projection has been suggested

in [11] without proof of its reliability (i.e., that the corresponding block Krylov matrix is non-singular). We establish here that it does yield a block Krylov matrix of full rank, and hence can be used for an efficient inverse of a sparse A .

Let $\mathcal{D} = \text{diag}(\delta_1, \dots, \delta_1, \delta_2, \dots, \delta_2, \dots, \delta_m, \dots, \delta_m)$ be an $n \times n$ diagonal matrix whose entries consist of m distinct indeterminates δ_i , each δ_i occurring s times.

THEOREM 2.1. *If the leading $ks \times ks$ minor of A is nonzero for $1 \leq k \leq m$, then $\mathcal{K}_m(\mathcal{D}AD, u) \in \mathbb{F}^{n \times n}$ is invertible.*

PROOF. Let $\mathcal{B} = \mathcal{D}AD$. For $1 \leq k \leq m$, define \mathcal{B}_k as the specialization of \mathcal{B} obtained by setting $\delta_{k+1}, \delta_{k+2}, \dots, \delta_m$ to zero. Then \mathcal{B}_k is the matrix constructed by setting to zero the last $n - ks$ rows and columns of \mathcal{B} . Similarly, for $1 \leq k \leq m$ we define $u_k \in \mathbb{F}^{n \times s}$ to be the matrix constructed from u by setting to zero the last $n - ks$ rows. In particular we have $\mathcal{B}_m = \mathcal{B}$ and $u_m = u$.

We proceed by induction on k and show that

$$\text{rank } \mathcal{K}_k(\mathcal{B}_k, u_k) = ks, \quad (2.2)$$

for $1 \leq k \leq m$. For the base case $k = 1$ we have $\mathcal{K}_1(\mathcal{B}_1, u_1) = u_1$ and thus $\text{rank } \mathcal{K}_1(\mathcal{B}_1, u_1) = \text{rank } u_1 = s$.

Now, assume that (2.2) holds for some k with $1 \leq k < m$. By the definition of \mathcal{B}_k and u_k , only the first ks rows of \mathcal{B}_k and u_k will be involved in the left hand side of (2.2). Similarly, only the first ks columns of \mathcal{B}_k will be involved. Since by assumption on \mathcal{B} the leading $ks \times ks$ minor is non-zero, we have $\text{rank } \mathcal{B}_k \mathcal{K}_k(\mathcal{B}_k, u_k) = ks$, which is equivalent to $\text{rank } \mathcal{K}_k(\mathcal{B}_k, \mathcal{B}_k u_k) = ks$. By the fact that the first ks rows of $u_{k+1} - u_k$ are zero, we have $\mathcal{B}_k(u_{k+1} - u_k) = 0$, or equivalently $\mathcal{B}_k u_{k+1} = \mathcal{B}_k u_k$, and hence

$$\text{rank } \mathcal{K}_k(\mathcal{B}_k, \mathcal{B}_k u_{k+1}) = ks. \quad (2.3)$$

The matrix in (2.3) can be written as

$$\mathcal{K}_k(\mathcal{B}_k, \mathcal{B}_k u_{k+1}) = \begin{bmatrix} M_k \\ 0 \end{bmatrix} \in \mathbb{F}^{n \times ks},$$

where $M_k \in \mathbb{F}^{ks \times ks}$ is nonsingular. Introducing the block u_{k+1} we obtain the following matrix:

$$[u_{k+1}, \mathcal{K}_k(\mathcal{B}_k, \mathcal{B}_k u_{k+1})] = \begin{bmatrix} * & M_k \\ I_s & 0 \\ 0 & 0 \end{bmatrix}. \quad (2.4)$$

whose rank is $(k+1)s$. Noticing that

$$\begin{bmatrix} u_{k+1}, \mathcal{K}_k(\mathcal{B}_k, \mathcal{B}_k u_{k+1}) \end{bmatrix} = \begin{bmatrix} \mathcal{K}_{k+1}(\mathcal{B}_k, u_{k+1}) \end{bmatrix},$$

we are led to

$$\text{rank } \mathcal{K}_{k+1}(\mathcal{B}_k, u_{k+1}) = (k+1)s.$$

Finally, using the fact that \mathcal{B}_k is the specialization of \mathcal{B}_{k+1} obtained by setting δ_{k+1} to zero, we obtain

$$\text{rank } \mathcal{K}_{k+1}(\mathcal{B}_{k+1}, u_{k+1}) = (k+1)s,$$

which is (2.2) for $k+1$ and thus establishes the theorem by induction. \square

If the leading $ks \times ks$ minor of A is nonzero then the leading $ks \times ks$ minor of A^T is nonzero as well, for any integer k .

COROLLARY 2.2. *If the leading $ks \times ks$ minor of A is nonzero for $1 \leq k \leq m$ and $\mathcal{B} = \mathcal{D}AD$, then $\mathcal{K}_m(\mathcal{B}^T, u) \in \mathbb{F}^{n \times n}$ is invertible.*

Suppose now that $A \in \mathbb{F}^{n \times n}$ is an arbitrary non-singular matrix and the size of \mathbb{F} exceeds $n(n+1)$. It follows by Theorem 2 of Kaltofen and Saunders [18] that there exists a lower triangular Toeplitz matrix $L \in \mathbb{F}^{n \times n}$ and an upper triangular Toeplitz matrix $U \in \mathbb{F}^{n \times n}$ such that each of the leading minors of $\hat{A} = UAL$ is nonzero. Let $\mathcal{B} = \mathcal{D}\hat{A}\mathcal{D}$; then the products of the determinants of the matrices $\mathcal{K}_m(\mathcal{B}, u)$ and $\mathcal{K}_m(\mathcal{B}^T, u)$ (mentioned in the above theorem and corollary) is a polynomial with total degree less than $2n(m-1) < n(n+1)$ (if $m \neq 1$). In this case it follows that there is also a non-singular diagonal matrix $D \in \mathbb{F}^{n \times n}$ such that $\mathcal{K}_m(B, u)$ and $\mathcal{K}_m(B^T, u)$ are non-singular, for

$$B = D\hat{A}D = DUALD.$$

Now let $R = LD^2U \in \mathbb{F}^{n \times n}$, $\hat{u} \in \mathbb{F}^{s \times n}$ and $\hat{v} \in \mathbb{F}^{n \times s}$ such that

$$\hat{u}^T = (L^T)^{-1}D^{-1}u \quad \text{and} \quad \hat{v} = LDu.$$

Then

$$\mathcal{K}_m(RA, \hat{v}) = LD\mathcal{K}_m(B, u)$$

and

$$L^T D\mathcal{K}_m((RA)^T, \hat{u}^T) = \mathcal{K}_m(B^T, u),$$

so that $\mathcal{K}_m(RA, \hat{v})$ and $\mathcal{K}_m((RA)^T, \hat{u}^T)$ are each non-singular as well. Since D is diagonal and U and L are triangular Toeplitz matrices it is now easily established that (R, \hat{u}, \hat{v}) is an “efficient block projection” for the given matrix A , where this is as defined in [11].

This proves Conjecture 2.1 of [11] for the case that the size of \mathbb{F} exceeds $n(n+1)$:

COROLLARY 2.3. *For any non-singular $A \in \mathbb{F}^{n \times n}$ and $s|n$ (over a field of size greater than $n(n+1)$) there exists an efficient block projection $(R, u, v) \in \mathbb{F}^{n \times n} \times \mathbb{F}^{s \times n} \times \mathbb{F}^{n \times s}$.*

3. FACTORIZATION OF THE MATRIX INVERSE

The existence of the efficient block projection established in the previous section allows us to define a useful factorization of the inverse of a matrix. This was used to obtain faster heuristics for solving sparse integer matrices in [11]. The basis is the following factorization of the matrix inverse.

Let $\mathcal{B} = \mathcal{D}AD$, where \mathcal{D} is an $n \times n$ diagonal matrix whose diagonal entries consist of m distinct indeterminates, each occurring s times contiguously, as previously. Define $\mathcal{K}_u^{(r)} = \mathcal{K}_m(\mathcal{B}, u)$ with u as in (2.1) and $\mathcal{K}_u^{(\ell)} = \mathcal{K}_m(\mathcal{B}^T, u)^T$ (where (r) and (ℓ) refer to projection on the right and left respectively). For any $0 \leq k \leq m-1$ and any two indices l and r such that $l+r=k$ we have $u^T \mathcal{B}^l \cdot \mathcal{B}^r u = u^T \mathcal{B}^k u$. Hence the matrix $\mathcal{H}_u = \mathcal{K}_u^{(\ell)} \cdot \mathcal{B} \cdot \mathcal{K}_u^{(r)}$ is block-Hankel with blocks of dimension $s \times s$:

$$\mathcal{H}_u = \begin{bmatrix} u^T \mathcal{B} u & u^T \mathcal{B}^2 u & \dots & u^T \mathcal{B}^m u \\ u^T \mathcal{B}^2 u & u^T \mathcal{B}^3 u & \ddots & \vdots \\ \vdots & \ddots & \ddots & u^T \mathcal{B}^{2m-2} u \\ u^T \mathcal{B}^m u & \dots & u^T \mathcal{B}^{2m-2} u & u^T \mathcal{B}^{2m-1} u \end{bmatrix} \in \mathbb{F}^{n \times n}$$

From $\mathcal{H}_u = \mathcal{K}_u^{(\ell)} \cdot \mathcal{B} \cdot \mathcal{K}_u^{(r)} = \mathcal{K}_u^{(\ell)} \cdot \mathcal{D}AD \cdot \mathcal{K}_u^{(r)}$, we have following corollary to Theorem 2.1, which (together with

Corollary 2.2) implies that $\mathcal{K}_u^{(r)}$ and $\mathcal{K}_u^{(\ell)}$, and thus \mathcal{H}_u are invertible.

COROLLARY 3.1. *If $A \in \mathbb{F}^{n \times n}$ is such that all leading $ks \times ks$ minors are non-singular, \mathcal{D} is a diagonal matrix of indeterminates and $\mathcal{B} = \mathcal{D}AD$, then \mathcal{B}^{-1} and A^{-1} may be factored as*

$$\begin{aligned} \mathcal{B}^{-1} &= \mathcal{K}_u^{(r)} \mathcal{H}_u^{-1} \mathcal{K}_u^{(\ell)}, \\ A^{-1} &= \mathcal{D} \mathcal{K}_u^{(r)} \mathcal{H}_u^{-1} \mathcal{K}_u^{(\ell)} \mathcal{D}, \end{aligned} \quad (3.1)$$

where $\mathcal{K}_u^{(\ell)}$ and $\mathcal{K}_u^{(r)}$ are block-Krylov as defined above, and $\mathcal{H}_u \in \mathbb{F}^{n \times n}$ is block-Hankel (and invertible) with $s \times s$ blocks, as above.

We note that for any specialization of the indeterminates in \mathcal{D} to field elements in \mathbb{F} such that $\det \mathcal{H}_u \neq 0$ we get a similar formula to (3.1) completely over \mathbb{F} . A similar factorization in the non-blocked case is used in [10, (4.5)] for fast parallel matrix inversion.

4. BLACK-BOX MATRIX INVERSION OVER A FIELD

Let $A \in \mathbb{F}^{n \times n}$ be invertible and such that for any $v \in \mathbb{F}^{n \times 1}$ the matrix times vector product Av or $A^T v$ can be computed in $\phi(n)$ operations in \mathbb{F} (where $\phi(n) \geq n$). Following Kaltofen, we call such matrix-vector and vector-matrix products *black-box* evaluations of A . In this section we will show how to compute A^{-1} with $O(n^{2-1/(\omega-1)})$ black box evaluations and additional $O(n^{3-1/(\omega-1)})$ operations in \mathbb{F} . Note that when $\phi(n) = O(n)$ — a common characterization of “sparse” matrices — the exponent in n of this cost is smaller than ω , and is $O(n^{2.273})$ with the currently best-known matrix multiplication.

We assume for the moment that all principal $ks \times ks$ minors of A are non-zero, $1 \leq k \leq m$.

Let $\delta_1, \delta_2, \dots, \delta_m$ be the indeterminates which form the diagonal entries of \mathcal{D} and let $\mathcal{B} = \mathcal{D}AD$. By Theorem 2.1 and Corollary 2.2, the matrices $\mathcal{K}_m(\mathcal{B}, u)$ and $\mathcal{K}_m(\mathcal{B}^T, u)$ are each invertible. If $m \geq 2$ then the product of the determinants of these matrices is a non-zero polynomial $\Delta \in \mathbb{F}[\delta_1, \dots, \delta_m]$ with total degree less than $2n(m-1) - 1$.

Suppose that \mathbb{F} has at least $2n(m-1)$ elements. Then Δ cannot be zero at all points in $(\mathbb{F} \setminus \{0\})^n$. Let d_1, d_2, \dots, d_m be nonzero elements of \mathbb{F} such that $\Delta(d_1, d_2, \dots, d_m) \neq 0$, let $D = \text{diag}(d_1, \dots, d_1, \dots, d_m, \dots, d_m)$, and let $B = \mathcal{D}AD$. Then $K_u^{(r)} = \mathcal{K}_m(B, u) \in \mathbb{F}^{n \times n}$ and $K_u^{(\ell)} = \mathcal{K}_m(B^T, u)^T \in \mathbb{F}^{n \times n}$ are each invertible since $\Delta(d_1, d_2, \dots, d_m) \neq 0$, B is invertible since A is and d_1, d_2, \dots, d_m are all nonzero, and thus $H_u = K_u^{(\ell)} B K_u^{(r)} \in \mathbb{F}^{n \times n}$ is invertible as well. Correspondingly, (3.1) suggests

$$B^{-1} = K_u^{(r)} H_u^{-1} K_u^{(\ell)}, \quad \text{and} \quad A^{-1} = D K_u^{(r)} H_u^{-1} K_u^{(\ell)} D$$

for computing the matrix inverse.

1. Computation of $u^T, u^T B, \dots, u^T B^{2m-1}$ and $K_u^{(\ell)}$.

We can compute this sequence, and hence $K_u^{(r)}$ with $m-1$ applications of B to vectors using $O(n\phi(n))$ operations in \mathbb{F} .

2. Computation of H_u .

Due to the special form (2.1) of u , one may then compute wu for any $w \in \mathbb{F}^{s \times n}$ with $O(sn)$ operations.

Hence we can now compute $u^T B^i u$ for $0 \leq i \leq 2m-1$ with $O(n^2)$ operations in \mathbb{F} .

3. Computation of H_u^{-1} .

The off-diagonal inverse representation of H_u^{-1} as in (A.4) in the Appendix can be found with $O(s^\omega m)$ operations by Proposition A.1.

4. Computation of $H_u^{-1} K_u^{(\ell)}$.

From Corollary A.2 in the Appendix, we can compute the product $H_u^{-1} M$ for any matrix $M \in \mathbb{F}^{n \times n}$ with $O(s^\omega m^2)$ operations.

5. Computation of $K_u^{(r)} \cdot (H_u^{-1} K_u^{(\ell)})$.

We can compute $K_u^{(r)} M = [u, Bu, \dots, B^{m-1}u]M$, for any $M \in \mathbb{F}^{n \times n}$ by splitting M into m blocks of s consecutive rows M_i , for $0 \leq i \leq m-1$:

$$\begin{aligned} K_u M &= \sum_{i=0}^{m-1} B^i (u M_i) \\ &= u M_0 + B(u M_1 + B(u M_2 + \dots \\ &\quad \dots + B(u M_{m-2} + B(u M_{m-1}) \dots)). \end{aligned} \quad (4.1)$$

Because of the special form (2.1) of u , each product $u M_i \in \mathbb{F}^{n \times n}$ requires $O(n^2)$ operations, and hence all such products involved in (4.1) can be computed in $O(mn^2)$ operations. Since applying B to an $n \times n$ matrix costs $n\phi(n)$ operations, $K_u^{(r)} M$ is computed in $O(mn\phi(n) + mn^2)$ operations using the iterative form of (4.1).

In total, the above process requires $O(mn)$ applications of A to a vector (the same as for B), and $O(s^\omega m^2 + mn^2)$ additional operations. If $\phi(n) = O(n)$ the overall number of field operations is minimized with the blocking factor $s = n^{1/(\omega-1)}$.

THEOREM 4.1. *Let $A \in \mathbb{F}^{n \times n}$, where $n = ms$, be such that all leading $ks \times ks$ minors are non-singular for $1 \leq k \leq m$. Let $B = DAD$ for $D = \text{diag}(d_1, \dots, d_1, \dots, d_m, \dots, d_m)$ such that d_1, d_2, \dots, d_m are nonzero and each of the matrices $\mathcal{K}_m(DAD, u)$ and $\mathcal{K}_m((DAD)^T, u)$ is invertible. Then the inverse matrix A^{-1} can be computed using $O(n^{2-1/(\omega-1)})$ applications of A to vectors and an additional $O(n^{3-1/(\omega-1)})$ operations in \mathbb{F} .*

The above discussion makes a number of assumptions.

First, it assumes that the blocking factor s exactly divides n . This is easily accommodated by simply extending n to the nearest multiple of s , placing A in the top left corner of the augmented matrix, and adding diagonal ones in the bottom right corner.

Theorem 4.1 also makes the assumptions that all the leading $ks \times ks$ minors of A are non-singular and that the determinants of $\mathcal{K}_m(DAD, u)$ and $\mathcal{K}_m((DAD)^T, u)$ are each nonzero. While we know of no way to ensure this deterministically in the times given, standard techniques can be used to obtain these properties probabilistically if \mathbb{F} is sufficiently large.

Suppose, in particular, that $n \geq 16$ and that $\#\mathbb{F} > 2(m+1)n\lceil \log_2 n \rceil$. Fix a set \mathcal{S} of at least $2(m+1)n\lceil \log_2 n \rceil$ nonzero elements of \mathbb{F} . We can ensure that the leading $ks \times ks$ minors of A are non-zero by pre-multiplying by a butterfly network preconditioner U with parameters chosen

uniformly and randomly from \mathcal{S} . If U is constructed using the generic exchange matrix of [5, §6.2], then it will use at most $n\lceil \log_2 n \rceil/2$ random elements from \mathcal{S} , and from [5, Theorem 6.3] it follows that all leading $ks \times ks$ minors of $\tilde{A} = UA$ will be non-zero simultaneously with probability at least $3/4$. This probability of success can be made arbitrarily close to 1 with a choice from a larger \mathcal{S} . We note that $A^{-1} = \tilde{A}^{-1}U$. Thus, once we have computed \tilde{A}^{-1} we can compute A^{-1} with an additional $O(n^2)$ operations in \mathbb{F} , using the fact that multiplication of an arbitrary $n \times n$ matrix by an $n \times n$ butterfly preconditioner can be done with $O(n^2)$ operations.

Once again let Δ be the products of the determinants of the matrices $\mathcal{K}_m(DAD, u)$ and $\mathcal{K}_m((DAD)^T, u)$, so that Δ is nonzero with total degree less than $2n(m-1)$. If we choose randomly selected values from \mathcal{S} for $\delta_1, \dots, \delta_m$, since $\#\mathcal{S} \geq 2(m+1)n\lceil \log_2 n \rceil > 4 \deg \Delta$, the probability that Δ is zero at this point is at most $1/4$ by the Schwartz-Zippel Lemma [22, 24].

In summary, for randomly selected butterfly preconditioner B , and independently and randomly chosen values d_1, d_2, \dots, d_m the probability that $\tilde{A} = UA$ has non-singular leading $ks \times ks$ minors for $1 \leq k \leq m$ and $\Delta(d_1, d_2, \dots, d_m)$ is non-zero is at least $9/16 > 1/2$ when random choices are made uniformly and independently from a finite subset \mathcal{S} of $\mathbb{F} \setminus \{0\}$ with size at least $2(m+1)n\lceil \log_2 n \rceil$.

When $\#\mathbb{F} \leq 2(m+1)n\lceil \log_2 n \rceil$ we can easily construct a field extension \mathbb{E} of \mathbb{F} which has size greater than $2(m+1)n\lceil \log_2 n \rceil$ and perform the computation in that extension. Since this extension will have degree $O(\log_{\#\mathbb{F}} n)$ over \mathbb{F} , it will add only a logarithmic factor to the final cost. While we certainly do not claim that this is not of practical concern, it does not affect the asymptotic complexity.

Finally, we note that it is easily checked whether the matrix returned by this algorithm is the inverse of the input by using n multiplications by A by the columns of the output matrix and corresponding each to the corresponding column of the identity matrix. This results in a Las Vegas algorithm for computation of the inverse of a black-box matrix with the cost as given above.

THEOREM 4.2. *Let $A \in \mathbb{F}^{n \times n}$ be non-singular. Then the inverse matrix A^{-1} can be computed by a Las Vegas algorithm whose expected cost is $O(n^{2-1/(\omega-1)})$ applications of A to a vector and $O(n^{3-1/(\omega-1)})$ additional operations in \mathbb{F} .*

ω	Black-box applications	Blocking factor s	Inversion cost
3 (Standard)	1.5	1/2	$O(n^{2.5})$
2.807 (Strassen)	1.446	0.553	$O(n^{2.446})$
2.3755 (Cop/Win)	1.273	0.728	$O(n^{2.273})$

Table 4.1: Exponents of matrix inversion with a matrix \times vector cost $\phi(n) = O(n)$.

REMARK 4.3. *The structure (2.1) of the projection u plays a central role in computing the product of the block Krylov matrix by a $n \times n$ matrix. For a general projection $u \in \mathbb{F}^{n \times s}$, how to do better than a general matrix multiplication, i.e.,*

how to take advantage of the Krylov structure for computing $K_u M$, appears to be unknown.

Applying a Black-Box Matrix Inverse to a Matrix

The above method can also be used to compute $A^{-1}M$ for any matrix $M \in \mathbb{F}^{n \times n}$ with the same cost as in Theorem 4.2. Consider the new step 1.5:

1.5. Computation of $K_u^{(\ell)} \cdot M$.

Split M into m blocks of s columns, so that $M = [M_0, \dots, M_{m-1}]$ where $M_k \in \mathbb{F}^{n \times s}$. Now consider computing $K_u^{(\ell)} \cdot M_k$ for some $k \in \{0, \dots, m-1\}$. This can be accomplished by computing $B^i M_k$ for $0 \leq i \leq m-1$ in sequence, and then multiplying on the left by u^T to compute $u^T B^i M_k$ for each iterate.

The cost for computing $K_u^{(\ell)} M_k$ for a single k by the above process is $n - s$ multiplication of A to vectors and $O(ns)$ additional operations in \mathbb{F} . The cost of doing this for all k such that $0 \leq k \leq m-1$ is thus $m(n-s) < nm$ multiplications of A to vectors and $O(n^2)$ additional operations. Since applying A (and hence B) to an $n \times n$ matrix is assumed to cost $n\phi(n)$ operations in \mathbb{F} , $K_u^{(\ell)} \cdot M$ is computed in $O(mn\phi(n) + mn^2)$ operations in \mathbb{F} by the process described here.

Note that this is the same as the cost of Step 5, so the overall cost estimate is not affected. Since Step 4 does not rely on any special form for $K_u^{(\ell)}$, we can replace it with a computation of $H_u^{-1} \cdot (K_u^{(\ell)} M)$ with the same cost. We obtain the following:

COROLLARY 4.4. *Let $A \in \mathbb{F}^{n \times n}$ be non-singular and let $M \in \mathbb{F}^{n \times n}$ be any matrix. We can compute $A^{-1}M$ using a Las Vegas algorithm whose expected cost is $O(n^{2-1/(\omega-1)})$ multiplications of A to vectors and $O(n^{3-1/(\omega-1)})$ additional operations in \mathbb{F} .*

The estimates in Table 4 apply to this computation as well.

5. APPLICATIONS TO BLACK-BOX MATRICES OVER A FIELD

The algorithms of the previous section have applications in some important computations with black-box matrices over an arbitrary field \mathbb{F} . In particular, we consider the problems of computing the nullspace and rank of a black-box matrix. Each of these algorithms is probabilistic of the Las Vegas type. That is, the output is certified to be correct.

Kaltofen & Saunders [18] present algorithms for computing the rank of a matrix and for randomly sampling the nullspace, building upon work of Wiedemann [23]. In particular, they show that for random lower upper and lower triangular Toeplitz matrices $U, L \in \mathbb{F}^{n \times n}$, and random diagonal D , that all leading $k \times k$ minors of $\tilde{A} = UALD$ are non-singular for $1 \leq k \leq r = \text{rank } A$, and that if $f^{\tilde{A}} \in \mathbb{F}[x]$ is the minimal polynomial of \tilde{A} , then it has degree $r+1$ if A is singular (and degree n if A is non-singular). This is proven to be true for any input $A \in \mathbb{F}^{n \times n}$, and for random choice of U, L and D , with high probability. The cost of computing $f^{\tilde{A}}$ (and hence $\text{rank } A$) is shown to be $O(n)$ applications of

the black-box for A and $O(n^2)$ additional operations in \mathbb{F} . However, no certificate is provided that the rank is correct within this cost (and we do not know of one or provide one here). Kaltofen & Saunders [18] also show how to generate a vector uniformly and randomly from the nullspace of A with this cost (and, of course, this is certifiable with a single evaluation of the black box for A). We also note that the algorithms of Wiedemann and Kaltofen & Saunders require only a linear amount of extra space, which will not be the case for our algorithms.

We first employ the random preconditioning of [18]: $\tilde{A} = UALD$ as above. We will thus assume in what follows that A has all leading $i \times i$ minors non-singular for $1 \leq i \leq r$. While an unlucky choice may make this statement false, this case will be identified in our method. Also assume that we have computed the rank r of A with high probability. Again, this will be certified in what follows.

1. Inverting the leading minor.

Let A_0 be the leading $r \times r$ minor of A and partition A as

$$A = \begin{pmatrix} A_0 & A_1 \\ A_2 & A_3 \end{pmatrix}$$

Using the algorithm of the previous section, compute A_0^{-1} . If this fails, then the randomized conditioning or the rank estimate has failed and we either report this failure or try again with a different randomized preconditioning. If we can compute A_0^{-1} , then the rank of A is at least the estimated r .

2. Applying the inverted leading minor.

Compute $A_0^{-1}A_1 \in \mathbb{F}^{r \times (n-r)}$ using the algorithm of the previous section (this could in fact be merged into the first step).

3. Confirming the nullspace.

Note that

$$\begin{pmatrix} A_0 & A_1 \\ A_2 & A_3 \end{pmatrix} \underbrace{\begin{pmatrix} A_0^{-1}A_1 \\ -I \end{pmatrix}}_N = \begin{pmatrix} 0 \\ A_2A_0^{-1}A_1 - A_3 \end{pmatrix} = 0$$

and the Schur complement $A_2A_0^{-1}A_1 - A_3$ must be zero if the rank r is correct. This can be checked with $n-r$ evaluations of the black box for A . We note that because of its structure, N has rank $n-r$.

4. Output rank and nullspace basis.

If the Schur complement is zero, then output the rank r and N , whose columns give a basis for the nullspace of A . Otherwise, output fail (and possibly retry with a different randomized pre-conditioning).

THEOREM 5.1. *Let $A \in \mathbb{F}^{n \times n}$ have rank r . Then a basis for the nullspace of A and rank r of A can be computed with an expected number of $O(n^{2-1/(\omega-1)})$ applications of A to a vector, plus an additional expected number of $O(n^{3-1/(\omega-1)})$ operations in \mathbb{F} . The algorithm is probabilistic of the Las Vegas type.*

6. APPLICATIONS TO SPARSE RATIONAL LINEAR SYSTEMS

Given a non-singular $A \in \mathbb{Z}^{n \times n}$ and $b \in \mathbb{Z}^{n \times 1}$, in [11] we presented an algorithm and implementation to compute

$A^{-1}b$ with $O(n^{1.5}(\log(\|A\| + \|b\|)))$ matrix-vector products $v \mapsto Av \bmod p$ for a machine-word sized prime p and any $v \in \mathbb{Z}_p^{n \times 1}$ plus $O(n^{2.5}(\log(\|A\| + \|b\|)))$ additional bit-operations. Assuming that A and b had constant sized entries, and that a matrix-vector product by $A \bmod p$ could be performed with $O(n)$ operations modulo p , the algorithm presented could solve a system with $O(n^{2.5})$ bit operations. Unfortunately, this result was conditional upon the unproven Conjecture 2.1 of [11]: the existence of an efficient block projection. This conjecture was established in Corollary 2.3 of the current paper. We can now unconditionally state the following:

THEOREM 6.1. *Given any invertible $A \in \mathbb{Z}^{n \times n}$ and $b \in \mathbb{Z}^{n \times 1}$, we can compute $A^{-1}b$ using a Las Vegas algorithm. The expected number of matrix-vector products $v \mapsto Av \bmod p$ is in $O(n^{1.5}(\log(\|A\| + \|b\|)))$, and the expected number of additional bit-operations used by this algorithm is in $O(n^{2.5}(\log(\|A\| + \|b\|)))$.*

Sparse integer determinant and Smith form

The efficient block projection of Theorem 2.1 can also be employed relatively directly into the block baby-steps/giant-steps methods of [20] for computing the determinant of an integer matrix. This will yield improved algorithms for the determinant and Smith form of a sparse integer matrix. Unfortunately, the new techniques do not obviously improve the asymptotic cost of their algorithms in the case for which they were designed, namely, for computations of the determinants of dense integer matrices.

We only sketch the method for computing the determinant here following the algorithm in Section 4 of [20], and estimate its complexity. Throughout we assume that $A \in \mathbb{Z}^{n \times n}$ is non-singular and assume that we can compute $v \mapsto Av$ with $\phi(n)$ integer operations, where the bit-lengths of these integers are bounded by $O(\log(n + \|v\| + \|A\|))$.

Preconditioning and setup.

Precondition $A \leftarrow B = D_1 U A D_2$, where D_1, D_2 are random diagonal matrices, and U is a unimodular preconditioner from [23, §5]. While we will not do the detailed analysis here, selecting coefficients for these randomly from a set S_1 of size n^3 is sufficient to ensure a high probability of success. This preconditioning will ensure that all leading minors are non-singular and that the characteristic polynomial is squarefree with high probability (see [5] Theorem 4.3 for a proof of the latter condition). From Theorem 2.1, we also see that $\mathcal{K}_m(B, u)$ has full rank with high probability.

Let p be a prime which is larger than the a priori bound on the coefficients of the characteristic polynomial of A ; this is easily determined to be $(n \log \|A\|)^{n+o(1)}$. Fix a blocking factor s to be optimized later, and assume $n = ms$.

Choosing projections. Let $u \in \mathbb{Z}^{n \times s}$ be an efficient block projection as in (2.1) and $v \in \mathbb{Z}^{n \times s}$ a random (dense) block projection with coefficients chosen from a set S_2 of size at least $2n^2$.

Forming the sequence $\alpha_i = uA^i v \in \mathbb{Z}^{s \times s}$.

Compute this sequence for $i = 0 \dots 2m$. Computing all the $A^i v$ takes $O(n\phi(n) \cdot m \log \|A\|)$ bit operations. Computing all the $uA^i v$ takes $O(n^2 \cdot m \log \|A\|)$ bit operations.

Computing the minimal matrix generator.

The minimal matrix generator $F(\lambda)$ modulo p can be computed from the initial sequence segment $\alpha_0, \dots, \alpha_{2m-1}$. See [20, §4]. This can be accomplished with $O(ms^\omega \cdot n \log \|A\|)$ bit operations.

Extracting the determinant.

Following the algorithm in [20, §4], we first check if its degree is less than n and if so, return “failure”. Otherwise, we know $\det F^A(\lambda) = \det(\lambda I - A)$. Return $\det A = \det F(0) \bmod p$.

The correctness of the algorithm, and specifically the block projections, follows from fact that $[u, Au, \dots, A^{m-1}u]$ is of full rank with high probability by Theorem 2.1. Since the projection v is dense, the analysis of [20, (2.6)] is applicable, and the minimal generating polynomial will have full degree m with high probability, and hence its determinant at $\lambda = 0$ will be the determinant of A .

The total cost is $O((n\phi(n)m + n^2m + nms^\omega) \log \|A\|)$ bit operations, which is minimized when $s = n^{1/\omega}$. This yields an algorithm for the determinant which requires $O((n^{2-1/\omega}\phi(n) + n^{3-1/\omega}) \log \|A\|)$ bit operations. This is probably most interesting when $\omega = 3$, where it yields an algorithm for determinant which requires $O(n^{2.66} \log \|A\|)$ bit operations on a matrix with pseudo-linear cost matrix-vector product.

We also note that a similar approach allows us to use the Monte Carlo Smith form algorithm of [14], which is computed by means of computing the characteristic polynomial of random preconditionings of a matrix. This reduction is explored in [20] in the dense matrix setting. The upshot is that we obtain the Smith form with the same order of complexity, to within a poly-logarithmic factor, as we have obtained the determinant using the above techniques. See [20, §7.1] and [14] for details. We make no claim that this is practical in its present form.

APPENDIX

A. APPLYING THE INVERSE OF A BLOCK-HANKEL MATRIX

In this appendix we address asymptotically fast techniques for computing a representation of the inverse of a block Hankel matrix, for applying this inverse to an arbitrary matrix. The fundamental technique we will employ is to use the off-diagonal inversion formula of Beckermann & Labahn [1] and its fast variants [15]. An alternative to using the inversion formula would be to use the generalization of the Levinson-Durbin algorithm in [19].

For an integer m that divides n exactly with $s = n/m$, let

$$H = \begin{bmatrix} \alpha_0 & \alpha_1 & \dots & \alpha_{m-1} \\ \alpha_1 & \alpha_2 & \ddots & \vdots \\ \vdots & & \ddots & \alpha_{2m-2} \\ \alpha_{m-1} & \dots & \alpha_{2m-2} & \alpha_{2m-1} \end{bmatrix} \in \mathbb{F}^{n \times n} \quad (\text{A.1})$$

be a non-singular block-Hankel matrix whose blocks are $s \times s$ matrices over \mathbb{F} , and let α_{2m} be arbitrary in $\mathbb{F}^{s \times s}$. We follow the lines of [21] for computing the inverse matrix H^{-1} . Since H is invertible, the following four linear systems (see [21,

(3.8)-(3.11))

$$\begin{aligned} H[q_{m-1}, \dots, q_0]^t &= [0, \dots, 0, I] \in \mathbb{F}^{n \times s}, \\ H[v_m, \dots, v_1]^t &= -[\alpha_m, \dots, \alpha_{2m-1}, \alpha_{2m}] \in \mathbb{F}^{n \times s}, \end{aligned} \quad (\text{A.2})$$

and

$$\begin{aligned} [q_{m-1}^* \dots q_0^*] H &= [0 \dots 0 \ I] \in \mathbb{F}^{s \times n}, \\ [v_m^* \dots v_1^*] H &= -[\alpha_m \dots \alpha_{2m-1} \ \alpha_{2m}] \in \mathbb{F}^{s \times n}, \end{aligned} \quad (\text{A.3})$$

have unique solutions given by the $q_k, q_k^* \in \mathbb{F}^{s \times s}$, (for $0 \leq k \leq m-1$), and the $v_k, v_k^* \in \mathbb{F}^{s \times s}$ (for $1 \leq k \leq m$). Then we have (see [21, Theorem 3.1]):

$$\begin{aligned} H^{-1} &= \begin{bmatrix} v_{m-1} & \dots & v_1 & I \\ \vdots & \ddots & \vdots & \vdots \\ v_1 & \dots & \vdots & \vdots \\ I & \dots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} q_{m-1}^* & \dots & q_0^* \\ \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots \\ q_{m-1}^* & \dots & q_0^* \end{bmatrix} \\ &\quad - \begin{bmatrix} q_{m-2} & \dots & q_0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ q_0 & \dots & \vdots & \vdots \\ 0 & \dots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} v_m^* & \dots & v_1^* \\ \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots \\ v_m^* & \dots & v_1^* \end{bmatrix}. \end{aligned} \quad (\text{A.4})$$

The linear systems (A.2) and (A.3) may also be formulated in terms of matrix Padé approximation problems. We associate to H the matrix polynomial $A = \sum_{i=0}^{2m} \alpha_i x^i \in \mathbb{F}^{s \times s}[x]$. The $s \times s$ matrix polynomials Q, P, Q^*, P^* in $\mathbb{F}^{s \times s}[x]$ that satisfy

$$\begin{aligned} A(x)Q(x) &\equiv P(x) + x^{2m-1} \pmod{x^{2m}}, \\ \text{where } \deg Q &\leq m-1 \text{ and } \deg P \leq m-2, \\ Q^*(x)A(x) &\equiv P^*(x) + x^{2m-1} \pmod{x^{2m}}, \\ \text{where } \deg Q^* &\leq m-1 \text{ and } \deg P^* \leq m-2 \end{aligned} \quad (\text{A.5})$$

are unique and provide the coefficients $Q = \sum_{i=0}^{m-1} q_i x^i$ and $Q^* = \sum_{i=0}^{m-1} q_i^* x^i$ for constructing H^{-1} using (A.4) (see [21, Theorem 3.1]). The notation “mod x^i ” for $i \geq 0$ denotes that the terms of degree i or higher are forgotten. The $s \times s$ matrix polynomials V, U, V^*, U^* in $\mathbb{F}^{s \times s}[x]$ that satisfy

$$\begin{aligned} A(x)V(x) &\equiv U(x) \pmod{x^{2m+1}}, \quad V(0) = I, \\ \text{where } \deg V &\leq m \text{ and } \deg U \leq m-1, \\ V^*(x)A(x) &\equiv U^*(x) \pmod{x^{2m+1}}, \quad V^*(0) = I, \\ \text{where } \deg Q^* &\leq m-1 \text{ and } \deg P^* \leq m-2, \end{aligned} \quad (\text{A.6})$$

are unique and provide the coefficients $V = 1 + \sum_{i=1}^m v_i x^i$ and $Q^* = 1 + \sum_{i=1}^m v_i^* x^i$ for (A.4).

Using the matrix Padé formulation, the matrices Q, Q^*, V , and V^* may be computed using the σ -basis algorithm in [2], or its fast counterpart in [15, §2.2] that uses fast matrix multiplication. For solving (A.5), the σ -basis algorithm with $\sigma = s(2m-1)$ solves

$$\begin{aligned} [A \quad -I] \begin{bmatrix} \overline{Q} \\ \overline{P} \end{bmatrix} &= R x^{2m-1} \pmod{x^{2m}}, \\ [\overline{Q}^* \quad \overline{P}^*] \begin{bmatrix} A \\ -I \end{bmatrix} &= R^* x^{2m-1} \pmod{x^{2m}}, \end{aligned}$$

with $\overline{Q}, \overline{P}, \overline{Q}^*, \overline{P}^* \in \mathbb{F}^{s \times s}[x]$ that satisfy the degree constraints $\deg \overline{Q} \leq m-1, \deg \overline{Q}^* \leq m-1$, and $\deg \overline{P} \leq m-2, \deg \overline{P}^* \leq m-2$. The residue matrices R and R^* in $\mathbb{F}^{s \times s}$ are non-singular, hence $\overline{Q}R^{-1}$ and $(R^*)^{-1}\overline{Q}^*$ are solutions \overline{Q} and \overline{Q}^* for applying the inversion formula (A.4). For (A.6), the σ -basis algorithm with $\sigma = s(2m+1)$ leads to

$$\begin{aligned} [A \quad -I] \begin{bmatrix} \overline{V} \\ \overline{U} \end{bmatrix} &= \pmod{x^{2m+1}}, \\ [\overline{V}^* \quad \overline{U}^*] \begin{bmatrix} A \\ -I \end{bmatrix} &= \pmod{x^{2m+1}} \end{aligned}$$

with $\deg \overline{V} \leq m, \deg \overline{V}^* \leq m$, and $\deg \overline{U} \leq m-1, \deg \overline{U}^* \leq m-1$. The constant terms $\overline{V}(0)$ and $\overline{V}^*(0)$ in $\mathbb{F}^{s \times s}$ are non-singular, hence $V = \overline{V}(\overline{V}(0))^{-1}$ and $V^* = (\overline{V}^*(0))^{-1}\overline{V}^*$ are solutions for applying (A.4). Using Theorem 2.4 in [15] together with the above material we get the following cost estimate.

PROPOSITION A.1. *Computing the expression (A.4) of the inverse of the block-Hankel matrix (A.1) reduces to multiplying matrix polynomials of degree $O(m)$ in $\mathbb{F}^{s \times s}$, and can be done with $O(s^\omega m)$ operations in \mathbb{F} .*

Multiplying a block triangular Toeplitz or Hankel matrix in $\mathbb{F}^{n \times n}$ with blocks of size $s \times s$ by a matrix in $\mathbb{F}^{n \times n}$ reduces to the product of two matrix polynomials of degree $O(m)$, and of dimensions $s \times s$ and $s \times n$. Using the fast algorithms in [4] or [3], such a $s \times s$ product can be done in $O(s^\omega m)$ operations. By splitting the $s \times n$ matrix into $s \times s$ blocks, the $s \times s$ by $s \times n$ product can thus be done in $O(m \times s^\omega m) = O(s^\omega m^2)$ operations.

For $n = s^\nu$ let $\omega(1, 1, \nu)$ be the exponent of the problem of $s \times s$ by $s \times n$ matrix multiplication over \mathbb{F} . The splitting considered just above of the $s \times n$ matrix into $s \times s$ blocks, corresponds to taking $\omega(1, 1, \nu) = \omega + \nu - 1 < \nu + 1.376$ ($\omega < 2.376$ due to [6]), with the total cost $O(s^{\omega(1, 1, \nu)} m) = O(s^\omega m^2)$. Depending on $\sigma \geq 1$, a slightly smaller bound than $\nu + 1.376$ for $\omega(1, 1, \nu)$ may be used due the matrix multiplication techniques specifically designed for rectangular matrices in [16]. This is true as soon as $\nu \geq 1.171$, and gives for example $\omega(1, 1, \nu) < \nu + 1.334$ for $\nu = 2$, i.e., for $s = \sqrt{n}$.

COROLLARY A.2. *Let H be the block-Hankel matrix of (A.1). If the representation (A.4) of H^{-1} is given then computing $H^{-1}M$ for an arbitrary $M \in \mathbb{F}^{n \times n}$ reduces to four $s \times s$ by $s \times n$ products of polynomial matrices of degree $O(m)$. This can be done with $O(s^{\omega(1, 1, \nu)} m)$ or $O(s^\omega m^2)$ operations in \mathbb{F} ($n = s^\nu = ms$).*

B. REFERENCES

- [1] B. Beckermann and G. Labahn. A uniform approach for the fast, reliable computation of matrix-type padé approximants. *SIAM J. Matrix Anal. Appl.*, 15:804–823, 1994.
- [2] Bernhard Beckermann and George Labahn. A uniform approach for the fast computation of matrix-type Padé approximants. *SIAM Journal on Matrix Analysis and Applications*, 15(3):804–823, July 1994.
- [3] Alin Bostan and Eric Schost. Polynomial evaluation and interpolation on special sets of points. *J. Complex.*, 21(4):420–446, 2005.

- [4] D. Cantor and E. Kaltofen. Fast multiplication of polynomials over arbitrary algebras. *Acta Informatica*, 28:693–701, 1991.
- [5] L. Chen, W. Eberly, E. Kaltofen, B. D. Saunders, W. J. Turner, and G. Villard. Efficient matrix preconditioners for black box linear algebra. *Linear Algebra and its Applications*, 343–344:119–146, 2002.
- [6] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *J. Symb. Comp.*, 9:251–280, 1990.
- [7] John D. Dixon. Exact solution of linear equations using p -adic expansions. *Numerische Mathematik*, 40:137–141, 1982.
- [8] Jean-Guillaume Dumas, Thierry Gautier, Mark Giesbrecht, Pascal Giorgi, Bradford Hovinen, Erich Kaltofen, B. David Saunders, Will J. Turner, and Gilles Villard. LinBox: A generic library for exact linear algebra. In Arjeh M. Cohen, Xiao-Shan Gao, and Nobuki Takayama, editors, *Proceedings of the 2002 International Congress of Mathematical Software, Beijing, China*, pages 40–50. World Scientific, August 2002.
- [9] Jean-Guillaume Dumas, B. David Saunders, and Gilles Villard. Integer smith form via the valence: experience with large sparse matrices from homology. In *ISSAC '00: Proceedings of the 2000 international symposium on Symbolic and algebraic computation*, pages 95–105, New York, NY, USA, 2000. ACM Press.
- [10] Wayne Eberly. Processor-efficient parallel matrix inversion over abstract fields: two extensions. In *Second Int'l Symp. on Parallel Symbolic Computation (PASCO'97)*, pages 38–45, New York, NY, USA, 1997. ACM Press.
- [11] Wayne Eberly, Mark Giesbrecht, Pascal Giorgi, Arne Storjohann, and Gilles Villard. Solving sparse rational linear systems. In *ISSAC '06: Proceedings of the 2006 international symposium on Symbolic and algebraic computation*, pages 63–70, New York, NY, USA, 2006. ACM Press.
- [12] Ioannis Z. Emiris and Victor Y. Pan. Improved algorithms for computing determinants and resultants. *J. Complex.*, 21(1):43–71, 2005.
- [13] M. Giesbrecht. Efficient parallel solution of sparse systems of linear diophantine equations. In *Proceedings of PASCO'97*, pages 1–10, 1997.
- [14] M. Giesbrecht. Fast computation of the smith form of a sparse integer matrix. *Computational Complexity*, 10(1):41–69, 2004.
- [15] Pascal Giorgi, Claude-Pierre Jeannerod, and Gilles Villard. On the complexity of polynomial matrix computations. In Rafael Sendra, editor, *Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation, Philadelphia, Pennsylvania, USA*, pages 135–142. ACM Press, New York, August 2003.
- [16] Xiaohan Huang and Victor Y. Pan. Fast rectangular matrix multiplication and applications. *J. Complex.*, 14(2):257–299, 1998.
- [17] E. Kaltofen. Analysis of Coppersmith's block Wiedemann algorithm for the parallel solution of sparse linear systems. *Mathematics of Computation*, 64(210):777–806, 1995.
- [18] E. Kaltofen and B. D. Saunders. On Wiedemann's method of solving sparse linear systems. In *Proc. AAECC-9*, volume 539 of *Springer Lecture Notes in Comp. Sci.*, 1991. 29–38.
- [19] Erich Kaltofen. Analysis of Coppersmith's block Wiedemann algorithm for the parallel solution of sparse linear systems. *Mathematics of Computation*, 64(210):777–806, April 1995.
- [20] Erich Kaltofen and Gilles Villard. On the complexity of computing determinants. *Computational Complexity*, 13(3-4):91–130, 2004.
- [21] George Labahn, Dong Koo Chio, and Stan Cabay. The inverses of block hankel and block toeplitz matrices. *SIAM J. Comput.*, 19(1):98–123, 1990.
- [22] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. Assoc. Computing Machinery*, 27:701–717, 1980.
- [23] Douglas H. Wiedemann. Solving sparse linear equations over finite fields. *IEEE Transactions on Information Theory*, 32(1):54–62, January 1986.
- [24] R. Zippel. Probabilistic algorithms for sparse polynomials. In *Proc. EUROSAM 79*, pages 216–226, Marseille, 1979.